M2 AIC, SIGNAL PROCESSING : IMAGE SUPER-RESOLUTION RECONSTRUCTION

Bertelli Florian, Ramine Hamidi

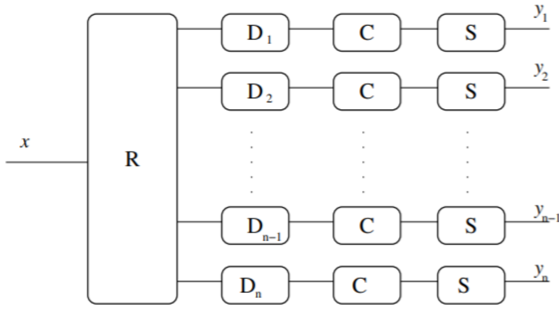
ABSTRACT

The aim of this paper is to present the project done during the signal processing course of the master 2 AIC, of the University Paris-Sud, with the teacher Matthieu Kowalski. The aim of this project is to implement super resolution, which permits to obtain from several low qualities images, an image with a better quality by combining the information from the several images mentionned before. First, we will from a single image simulate multiple low resolution and translated images, then reconstruct a good quality one from them. Finally, the aim will be to ”retrieve” the different translations and be able to reconstruct the image without precising the different translations, filters,etc... In this paper, we will use the famous image of Lena to implement, test, and try our algorithms.

The first part of this problem i.e the direct problem and the inverse of it will be present in the matlab publish : Direct Model. The second part deals with the true problem, i.e from a few images obtained from the direct model, but without supposing we know it, we want to generate an image with a higher resolution. In this part we will not assume anything about the original image, the convolution, the step or something else that we couldn't have in a real situation. This second part will be done in matlab publish Super Resolution.

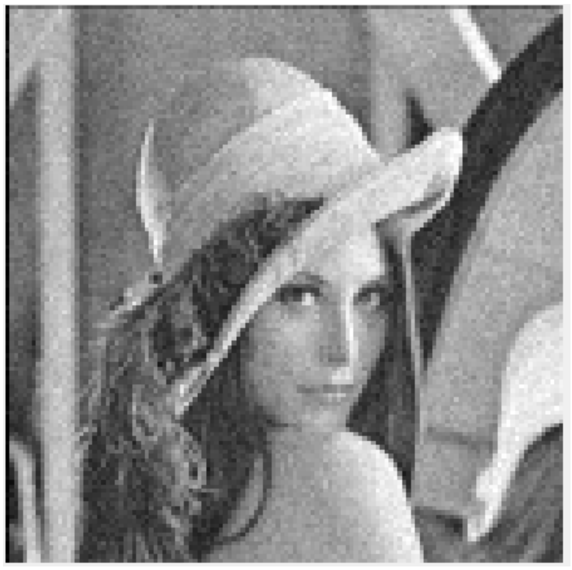
1. GENERATE THE DIRECT PROBLEM

As mentioned before, first we have to construct the direct problem. So, we want from a single picture to generate multiple translated and low resolution pictures. This process is well described by the figure below :



Direct model

Thus, with the initial lena's image we've generated 3 different images, by doing a translation, then a convolution, downsampling, and adding noise. For instance we've obtained the following image



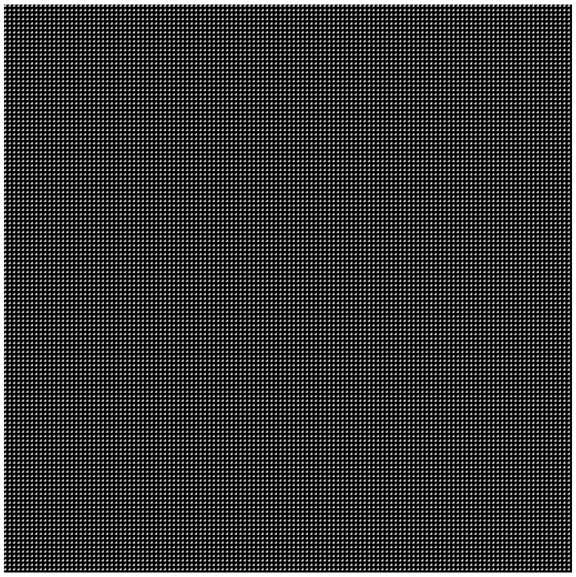
Example of results from the direct model

As we can see this image is clearly obtained by adding noise on the original image, sub sampled (1 pixel over 4), blurred with the convolution filter, and finally translated too (we can see the translation on the left of the image with a black line corresponding to the absence of pixel with the translation).

2. MASK AND RECONSTRUCTION

The goal is then to estimate the original high-resolution data x form its low resolution versions y_i that are several translated then sub-sampled version of x. For this, it is convenient to reformulate this problem as an inpainting problem. We can reformulate our problem, first by finding the z which is the image corresponding to the pixels of the low resolutions images, containing holes. Indeed, we want to find $z = M(x+e)$, where M is a binary mask of the size of the initial super resolution image, where a pixel is equal to one if we can obtained the corresponding pixel from the low resolution images, and zero otherwise. We get the following image z and the binary mask M.

Reconstructed image



Rreconstructed image's mask

We've computed the ratio of white "true pixels" ie the pixels obtained from the different low resolutions and we are dealing with only 18.75% of the filled pixels, the others are holes we need to fill in the next part.

3. INPAINTING

So now we've obtained an image of the original size, from lower resolution images and we still need to fill the holes. To fill these holes, which correspond to the absence of information we need to use a technique called inpainting. The aim of this technique is to find an estimation of the original signal x by minimizing a cost functional framework composed of a loss term :

$$L(x) = \frac{1}{2}||y - Ax||^2 \tag{1}$$

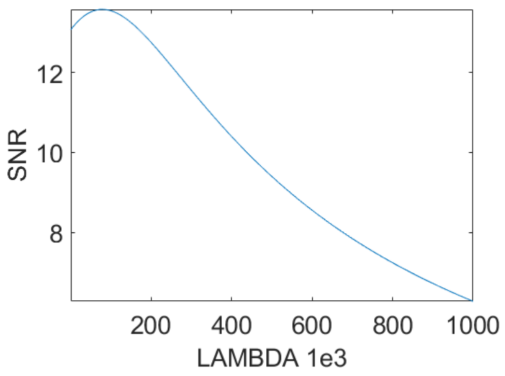
and a regularization term $R(x)$, also called prior, that aims to model the knowledge of the original signal x.

3.1. First method

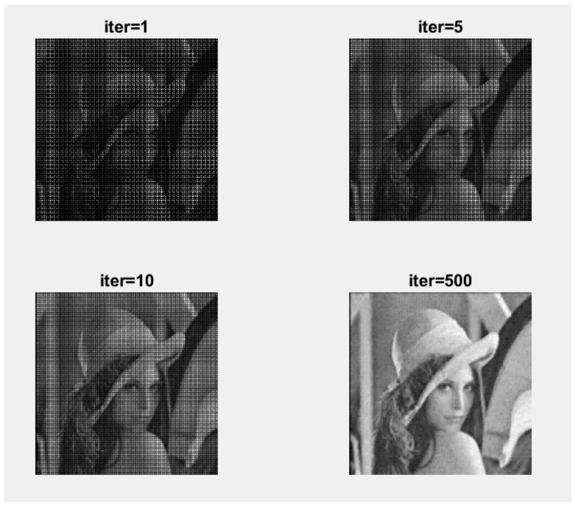
So our goal here is to implement a "descent" algorithm, ie an algorithm that finds $argmin(L(x) + \lambda R(x))$. To do this we have used an algorithm that takes the mask as input, and basically works on the holes, ie tries to fill them. We have used :

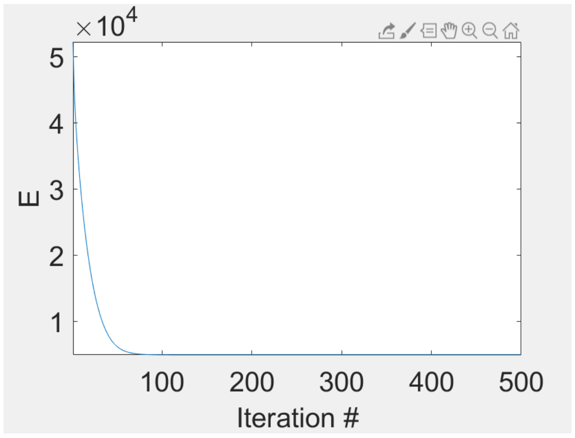
$$R(x) = ||\nabla^2 y||^2 \tag{2}$$

The technique we have used is called the Sobolev inpainting which is about finding $f^\star$ s.t $E(f) = \nabla f^2$ is minimal. Furthermore, after several tests we took a learning rate $\lambda = 0.0790$, because it appears to give the best tradeoff between the original signal and the obtained one. Indeed, we've tried for several λ and computed the SNR compared of the obtained image with the original image and we get :



So after we've chosen $\lambda = 0.0790$, we've applied the algorithm and there are evolution of the image and the power against the number of iterations.





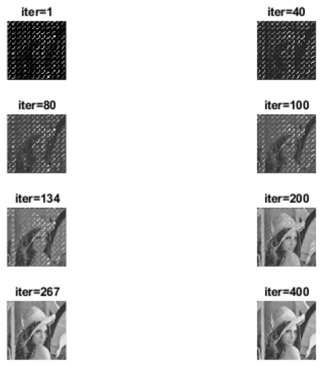
So with this we can clearly see that the Sobolev Energy is decreasing with the number of iterations, and we can see it visually on with the different images of Lena against the number of iterations. It appears that once we reach 100 iterations we reach the global maximum.

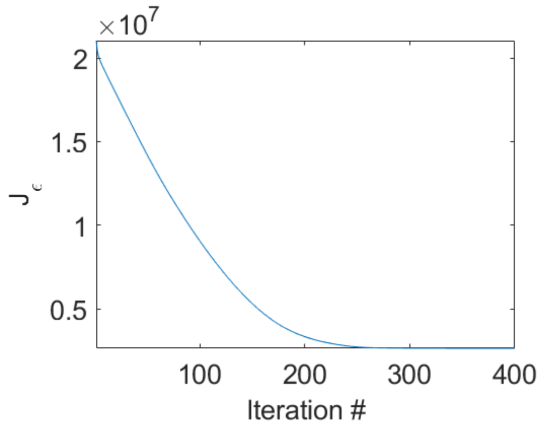
3.2. Second method

This method is present in the section ”filling the holes v2” of the first matlab publish. In this part we did inpainting by taking the prior

$$R(x) = \sum_x \sqrt{\nabla f(x)^2 + \epsilon^2}$$

We obtained the following results:





Visually we can see that the two methods work correctly, however in the super resolution section we will prefer the first one because it requires less iterations (50 against 200) to reach a minimum.

4. DECONVOLUTION

So now we have simulated the direct problem,and now we want to do the opposite, i.e how to obtain a blurry image from a high resolution image. Indeed, we have to perform, deconvolution, combination of the image, and impainting. So in this first step we are interested in the deconvolution step. In our direct model we have done it with a convolution with the following kernel :

$$h = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

This is basically low pass filter, which makes the image blurrier than the original one. So to inverse it we have done a kind of inverse filtering. The idea is to find a matrix which is the inverse of h and to apply it to the blurry image. We know that we can obtained the inverse of the initial convolution by taking the ”true” inverse of the Fourier transform of the low-pass filter. However, a problem is the Fourier transform of the low-pass filter have values close to zero or equal to it, so it will give us infinity. This is something we want to avoid, so we have decided to use a threshold, if the value of the low-pass filter is under it, we will put a coefficient chosen before and not the inverse.

We can clearly see that this algorithm works well, indeed the last image is quite the same as the original Lena, except it is blurrier. However, the essential difference is that now our image as a resolution of 512 x 512 against the original 128 x 128, so we are 4 times more precise which is very interesting and great.

5. TRANSLATION ESTIMATION

The aim of this part is to be able to estimate the translation if we don't know it, by using the correlation estimate. Indeed, in the real case we don't know at all the translations and we have to estimate it to realize the inverse problem. Here, basically what we do is to take a "reference image", one among the ones we have, and estimate the translations that were needed to obtain the other images. We did it in a "naive" way, which is that for each possible translation (quite few in reality because images are translated of a few pixels) compute the correlation and keep the translation with the highest correlation. We didn't obtained 100% of good answers with this computation of the correlation but however we've obtained translations that are near the true ones.

6. GENERATION OF HIGH RESOLUTION IMAGES

In this part our aim is to generate a higher resolution image from a few ones that have a low resolution. To simulate a real situation, we didn't have access to anything except the low resolution images. We used the previous part to determine the translation corresponding to each image, and then we produced an image of the size we want. In the following, there are the images obtained for different sizes, so different solutions, and we will be able to analyse the influence of the size of the final image against the quality. To make the comparison we will compute the percentage of filled pixel before inpainting.

Image obtained with 8.33% of filled pixel

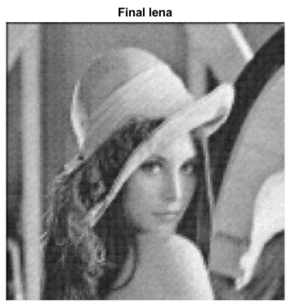


Image obtained with 18% of filled pixels

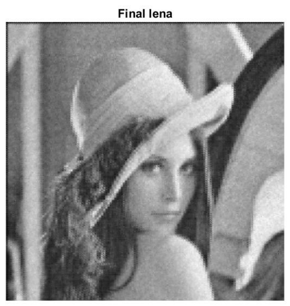
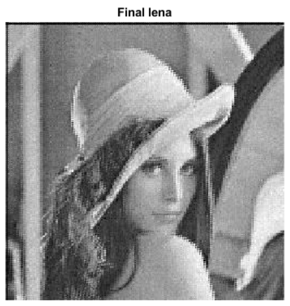
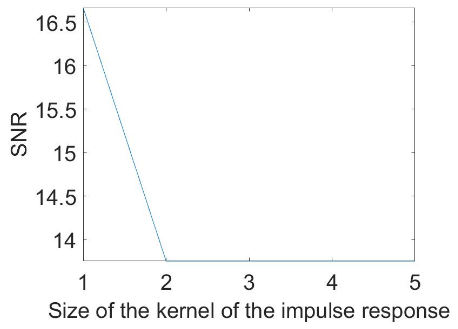


Image obtained with 87.5% of filled pixels



Finally we've tried with different impulse responses and compare the result with the SNR.

SNR against the size of the convolution kernel



We saw that the best solution is without any deconvolution, even if initially we've done it with a kernel of size 5.

Finally we obtain a SNR of 16.66 for the reconstructed picture, with only 3 pictures of 128x128 to reconstruct a picture 4 times bigger so with only initially 18.75% of the information. We believe that this is an interesting result because the SNR is high enough.